Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **4**, July **2016**

An Effective Mobile Visual Searching Algorithm Based on The Bag-of-Words Method for Furniture Images

Li Li, Dejie Zheng, Jianfeng Lu*

Institute of Graphics and Image, Hangzhou Dianzi University, Hangzhou, 310018
Zhejiang, China
*Corresponding author: jflu@hdu.edu.cn

Xiaoyang Mao

University of Yamanash, Japan

Chin-Chen Chang

Department of Information Engineering and Computer Science,
Feng Chia University, Taichung, 40724, Taiwan
alan3c@gmail.com

Received August, 2015; revised February, 2016

Abstract. *With the popularity of smart phones nowadays, image retrieval on mobile devices has been applied on a wider scale. To balance the efficiency and performance, an image retrieval algorithm which consists of two matching steps is proposed for furniture images captured by mobile devices (named mobile images). In the first stage, similar images are quickly obtained by the rough matching algorithm based on the bag-of-words (BOW) method. In the second stage, the most similar image is selected from the selection of similar images using the fine matching algorithm based on ORiented Brief (ORB).There are four main contributions in this paper. Firstly, add a spatial relationship to BOW features, an image is divided into three circular parts: the inner, middle and outer parts. Secondly, a feature point extraction method on concentric circles is proposed, which are resistant to geometric attack. Thirdly, the Gabor local line-based feature (GALIF) descriptor is improved by changing the filter object from the line to areas and overcome its weakness of low efficiency and rotation-sensitivity. Finally, a new image signature matching strategy is proposed based on Euclidean distance and a scale invariant feature to improve matching accuracy. Experimental results on a database with thousands of images show that our algorithm can retrieve mobile images quickly and precisely.*
Keywords: Three-party; Key agreement; Secret sharing; Security; Efficiency

1. **Introduction.** Nowadays, people are almost inseparable from their smart phones. People tend to get information from their surroundings by way of mobile images. This involves the question of mobile visual search. At present, there have been some mature applications on mobile visual search. In 2011, IKEA released their mobile application for their magazine for the iOS and android platforms. As shown in Figure 1, people can access the online shopping web page or some introductory videos after taking photos of the furniture images with their mobile phone. There are still a lot of similar applications, which involve mobile visual search technology.

The basic principle of image retrieval usually goes like this. Firstly, feature points are extracted from images, and then the feature points are described as a vector using some algorithm such as SIFT [1,]. Finally, the image feature vectors from the database are

FIGURE 1. IKEA's applications on mobile visual search

matched one by one and the image which has the highest correct matching rate will be regarded as the one most similar.

For the extraction of image feature points, the traditional method contains Harris corner detection [2,],features from Accelerated Segment Test (FAST)[3,], and so on. These methods can extract feature points accurately, but most of them are local image features and are time consuming.

For feature points description, SIFT [1,] and SURF[4,] et al. are widely used. The Gabor-based feature description method was first proposed by Gabor [5,] and has been widely used in the facial recognition field [6, 7,]. Eitz et al. [8,] proposed a Gabor local line-based feature (GALIF) description method. However, it has some shortcomings, such as rotation-sensitivity. Because Gabor filtering results are robust to lights, it is still very suitable for processing camera images.

With the increase of database capacity, the matching process of the feature point descriptors spends too much time on image retrieval. To speed up the process of image retrieval, Sivic [9,] referenced the method of text retrieval, called bag-of-words, which regards the feature descriptor as a visual word and the clustered feature descriptors as a visual vocabulary. With the aforementioned bag-of-words method, an image would be represented as the histogram of a word's frequency. The image retrieving process will be simplified to histogram matching. The performance of image retrieval surely can be improved, but the bag-of-words method abandons spatial information of an image thoroughly, which reduces the accuracy of image retrieval.

There exists a famous image matching method called ORB [10,] (oriented FAST and rotated BRIEF), which can match two feature descriptors quickly with a low cost of memory. The feature points are detected by oFAST [11,], which has been improved based on FAST [3,] and the feature is described by rBRIEF which has been improved

based on BRIEF [12,]. Considering the performance and the accuracy, we use ORB in our algorithm as the fine matching stage.

2. **The Overall Algorithm Flowchart.** In this section, we explain the overall algorithm flowchart, which is mainly divided into two parts; the database establishment and camera image retrieval stages.

(1)Database establishment stage: As shown in Figure 2, for every image in the database, we detect its image contour first using the Canny edge detector and by extracting the feature points by the concentric circles-based feature points extraction method. Next, the descriptor for feature points will be generated by the improved GALIF method. After that, feature points descriptors of all the images will be clustered into a visual vocabulary by the K-means clustering algorithm. Finally, the histogram of cluster occurrence generates a signature for the image. The visual vocabulary and signatures of all images will be preserved in a database for later use.

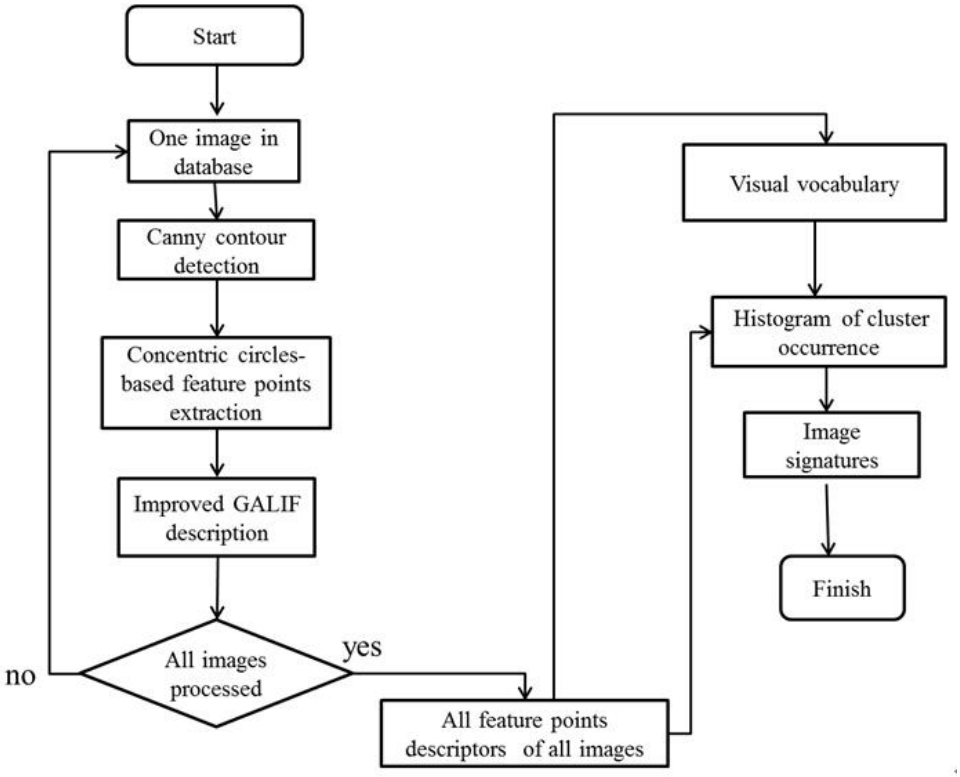


FIGURE 2. The database establishment stage

(2)Camera image retrieval stage: As shown in Figure 3, for a camera image, we firstly extract its feature points and describe them using the methods mentioned above. Then we generate its signature using the preserved visual vocabulary. After that, the signature will be matched with the preserved ones using our matching strategy and then a set of candidate images will be obtained at the result of the rough matching step. Finally, the most similar image will be found using the ORB method at the result of the fine matching step.

2.1. **The Novel Concentric Circles-based Feature Points Extraction Algorithm.** In order to speed up the image matching process, the feature points are extracted only on the contour of each image. Eitz et al.[8,] selected the feature points by evenly dividing the

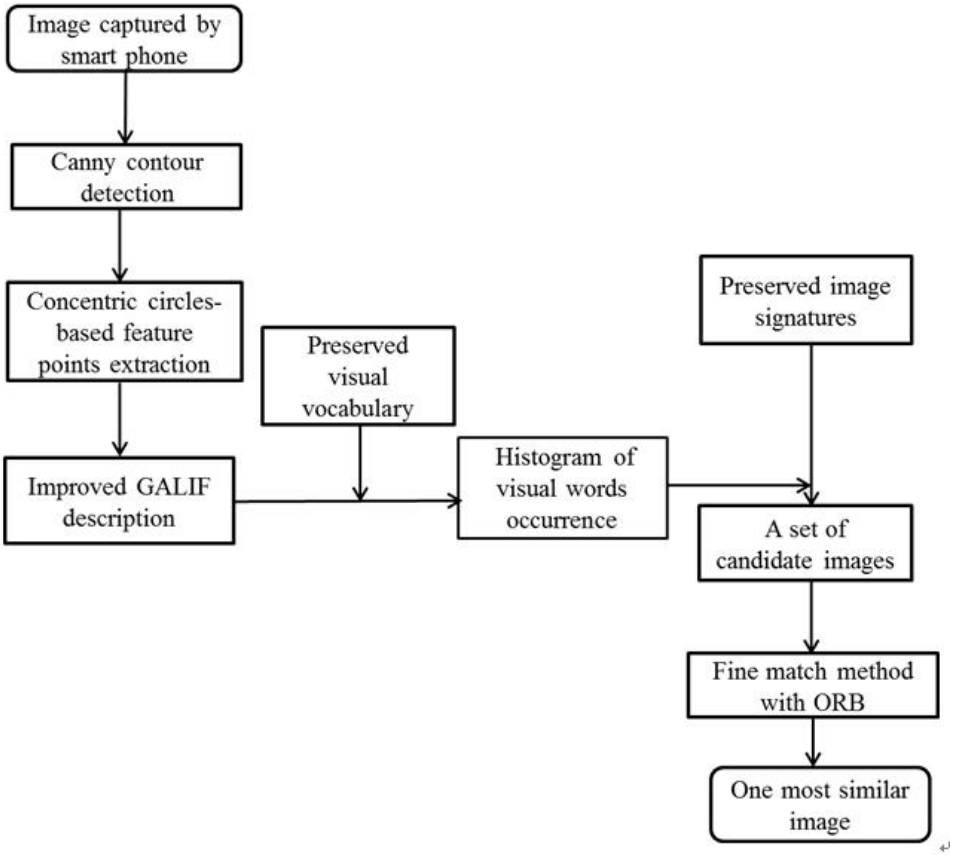
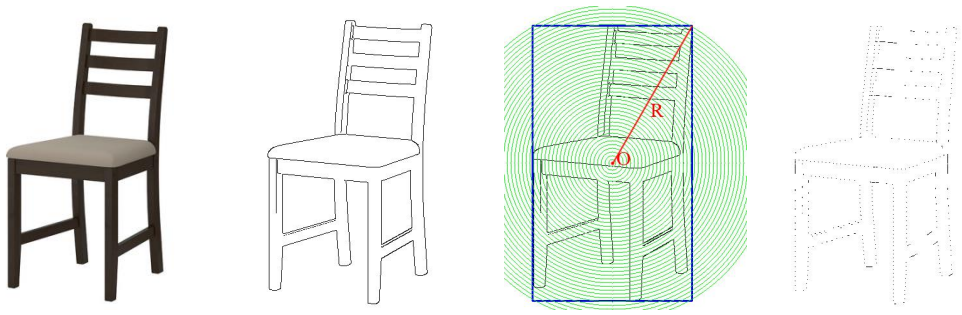


FIGURE 3. Camera image retrieval stage

image into 32×32 sub-blocks, and one point in each sub-block was selected as the feature point. Although this method is simple, it is not effective enough because the points selected are not evenly distributed throughout the useful part of the image. Sometimes there may not even be any contour points in the sub-block. Therefore, the selected points cannot reflect the characteristics of the contour and thus leads to mismatching. In order to solve this problem, we propose the concentric circles-based feature points selecting algorithm based on Canny boundary detection algorithm which is robust to geometric transformation. The steps are described as below:

(1)Image pre-processing. In order to ensure image consistency before and after photographing, the image pre-processing mainly includes the following: Due to cellphone cameras having high resolution, an image scaling adjustment is necessary which can reduce much process time on it. Affine transformation in camera images should be adjusted as well. To extract a better boundary contour, the photo image should also be sharpened. Figure 4(a) shows an original image whose contour image is extracted by Canny edge detector shown in Figure 4(b).

(2)Draw a bounding box on the contour image. The white pixels around the contour image should be removed as the background is not of interest to us. We define the half length of the bounding box' diagonal as R, as shown in Figure 4(c), and the feature points are selected in the bounding box area. By doing this, the processed area can be reduced, so the selecting speed of the feature points can be improved. Furthermore, this method can resist translation transformation because this method can ignore the relative position of interesting content in the image.



(a) original image (b) contour image (c) process of selecting points (d) feature points

FIGURE 4. The feature points selecting result

(3)Select points using concentric circles. The bounding box's center is defined as O and the interval distance between the adjacent concentric circles is defined as d. Draw concentric circles by the center of O and the first radius of r=d, and r increases by degree d until r=R, as shown in Figure 4(c). The interval distance d=αR, where α is the parameter defined to control the number of the concentric circles in this paper, being α=0.025. The points of intersection of concentric circles and the contour of the image are selected as the feature points p_i as shown in Figure 4(d). This feature points selecting method can resist image rotation because it is based on concentric circles. In addition, it is resistant to image resizing because the interval distance among the concentric circles changes based on parameter α.

Looking at Figure 4(d), we can see that the selecting feature points retain the shape of the original contour image well and the feature points are uniformly distributed.

Using our concentric circle-based feature points extraction algorithm, the feature points can selected evenly and robust to geometric attacks. What's more, the number of feature points can be regarded as the degree of density of the image, called density value. Before matching, we can use density value to distinguish the dense and the sparse image. When the difference of two density values is larger than a threshold, the two images will be regarded as different. Just by comparing the density values, the numbers of images in the database to match later will reduce greatly, which will save a great amount of time.

2.2. **Feature Descriptor Based on Improved GALIF.** The Gabor feature extraction method has been widely used in the visual information processing field. There are some reasons for using the Gabor feature extraction in our algorithm. First of all, Gabor features are not sensitive to changes in local lighting, which can be used to solve the illumination problems in a camera image. What's more, the Gabor filter is sensitive to the edge of the image, so its filtering results can reflect direction information of the gray gradation which can be used to describe the feature points extracted by the circle-based feature points extraction method.

A Gabor local line-based feature (GALIF) descriptor has been proposed by Eitz et al. [8]. In GALIF, the counter image was filtered by Gabor filter using 8 directions to describe a point. Although this works well, it has some shortcomings as well. Firstly, the filtering object of GALIF is a sketch image, so it just processes a subset of the lines in a sketch which doesn't take into account a lot of useful information. Considering the camera image, we change the filtering object to the gray image corresponding to the contours. Secondly, GALIF is not resistant to rotation. Finally, it is time consuming to filter the whole image. So a n×n patch around feature points is filtered which improves

TABLE 1. the effect of filtering with local patches

Feature	α	d	Patch size	T
100	0.025	9	5×5	1%
1000	0.05	18	9×9	3%
100	0.025	9	5×5	10%
1000	0.05	18	9×9	30%

the algorithm performance effectively in our method. The improved feature describing process is described as follows:

(1)For each feature point p_i extracted by the concentric circle-based feature points extraction method, we computes its coordinate direction with rectangle center O as y_i. The coordinate direction perpendicular to y_i is defined as x_i. As shown in Figure 5, x_i, y_i are regarded as the local coordinate system of p_i.

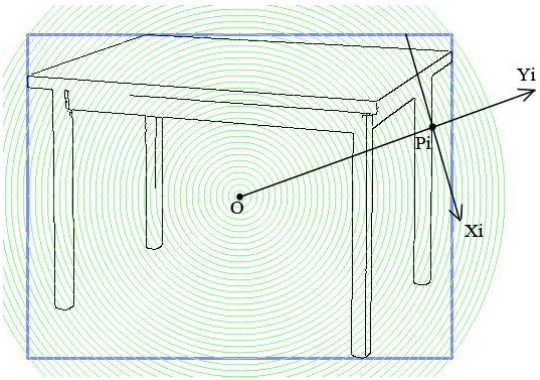


FIGURE 5. Local coordinate system of point p_i

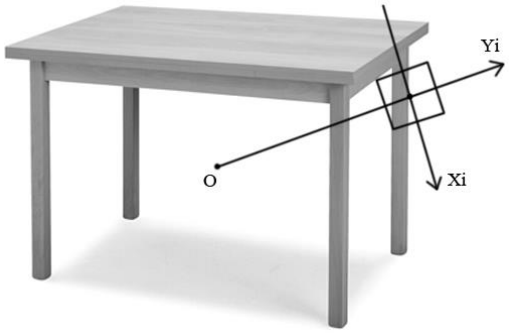


FIGURE 6. Local patch around point p_i

(2)Draw a n×n patch around p_i on the gray image corresponding to the contours, as shown in Figure 6. At the local coordinate system x_i, y_i divides the patch into four areas: A_i,i=1,2,3,4. To ensure that a descriptor contains characters of a patch sufficiently and is not disturbed by adjacent patches, we define that n=d/2, so the size of n is related to the interval distance of concentric circles.

(3)The local patch is filtered with Gabor filters in eight different directions at O_j,j=1,2,...,8. These directions are based on a local coordinate system and their values are among 0 and 180 degrees. We define a 32-dimensional vector F to store the filtering results. Because the filtering result of area A_i may contain more than one pixel, we define the mean value of the filtering result in area A_i as a_{ji} in the direction of o_j . The filtering results a_{ji} in 4 areas and 8 directions are assigned to vector F. Thus, a point will be regarded as 32-dimensional descriptor F, as shown in Figure 7.

Table 1 shows the effect of filtering with local patches. α and d are parameters mentioned in section 2.1. Parameter T is the ratio of actual filtering areas with local patches to filtering areas with the complete image. Table 1 shows that when feature numbers are among 100 and 1000, and T is just among 1% and 30%, the speed of filtering will clearly be improved.

We use the image in Figure 8 to test the anti-rotation of our algorithm, the original image is rotated by different. Table 2 shows the numbers of points extracted from Concentric Circles-based Feature Points Extraction Algorithm of each rotated image. The

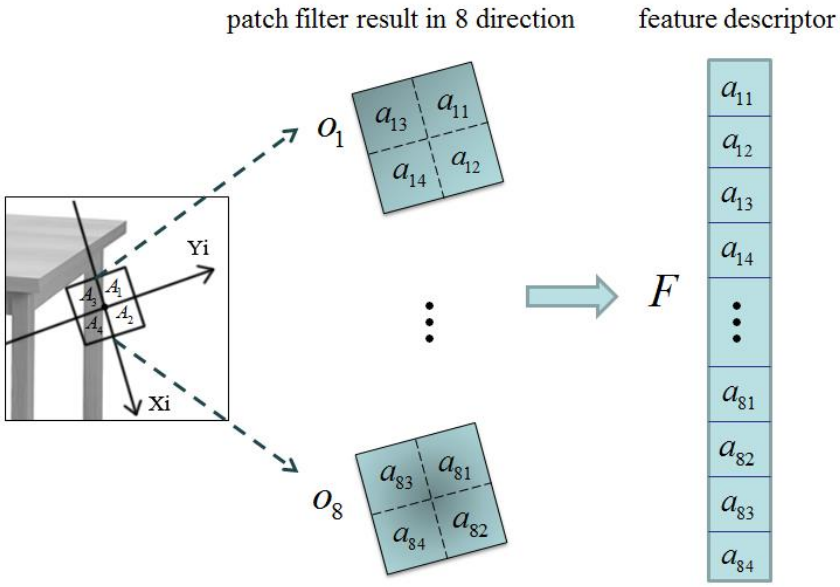


FIGURE 7. The process of improved GALIF

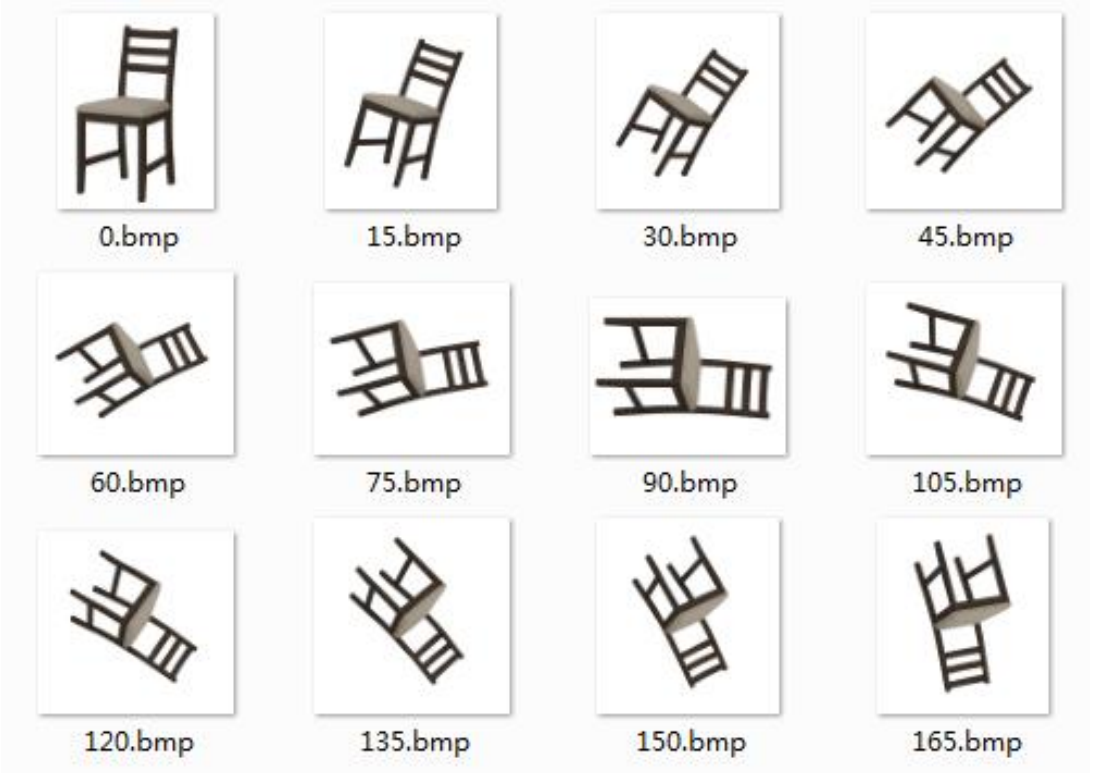


FIGURE 8. The rotated images

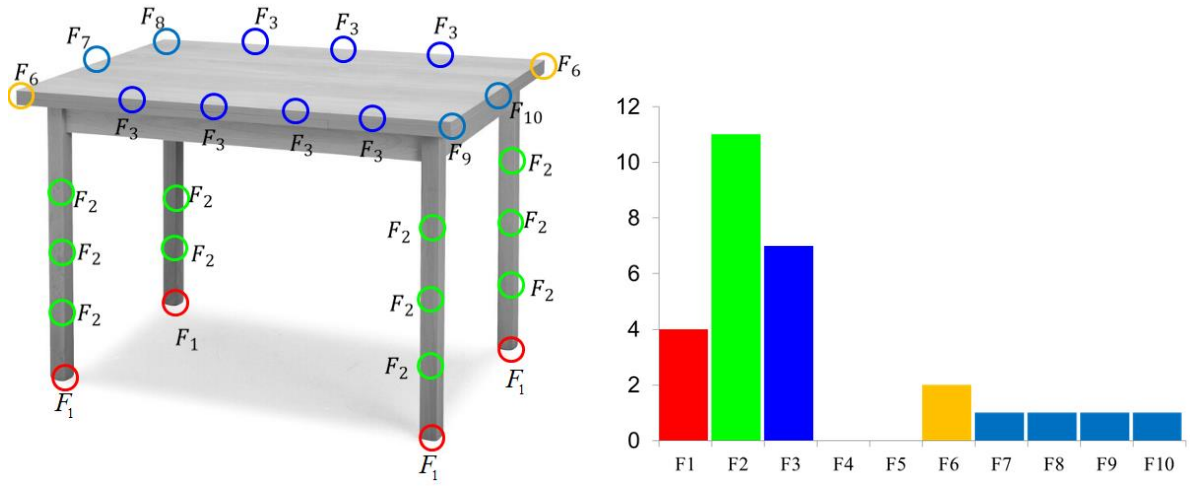
rotated images are described by our algorithm and their differences with the original image are measured by the Euclidean distance. It reveals that our algorithm with GALIF is anti-rotation.

2.3. **Image Signature Generation Method Based on Improved BOW.** After all the feature points descriptors of all images are stored in the database, a visual vocabulary using the K-means clustering method is obtained. Next, the histograms of cluster

TABLE 2. The numbers of points extracted

rotation	0	15	30	45	60	75	90	105	120	135	150	165
Feature Numbers	290	294	301	305	301	288	285	286	305	301	304	285
difference	0	0.0 171	0.0 124	0.0 200	0.0 135	0.0 109	0.0 039	0.0 119	0.0 028	0.0 147	0.0 081	0.0 192

occurrences generate signatures for the images in this way. By comparing each feature point descriptor in an image with visual words in the visual vocabulary, when finding the nearest word, the number of this word will increase. Continue until all the points feature descriptors are processed. Finally, a histogram of visual words occurrences will be produced as the signature of an image. For example, in Figure 9(a), different colored circles on the image represent different feature descriptors. Figure 9(b) shows the signature of the image. The horizontal coordinate represents a visual vocabulary which contains 10 words and the vertical coordinate represents the occurrence of visual word. So, the signature of the image is vector H, H=[4,11,7,0,0,2,1,1,1,1]. To compare with different images, the vector H will be normalized as a result.



(a) Feature points on the image (b) Histogram of visual words occurrence

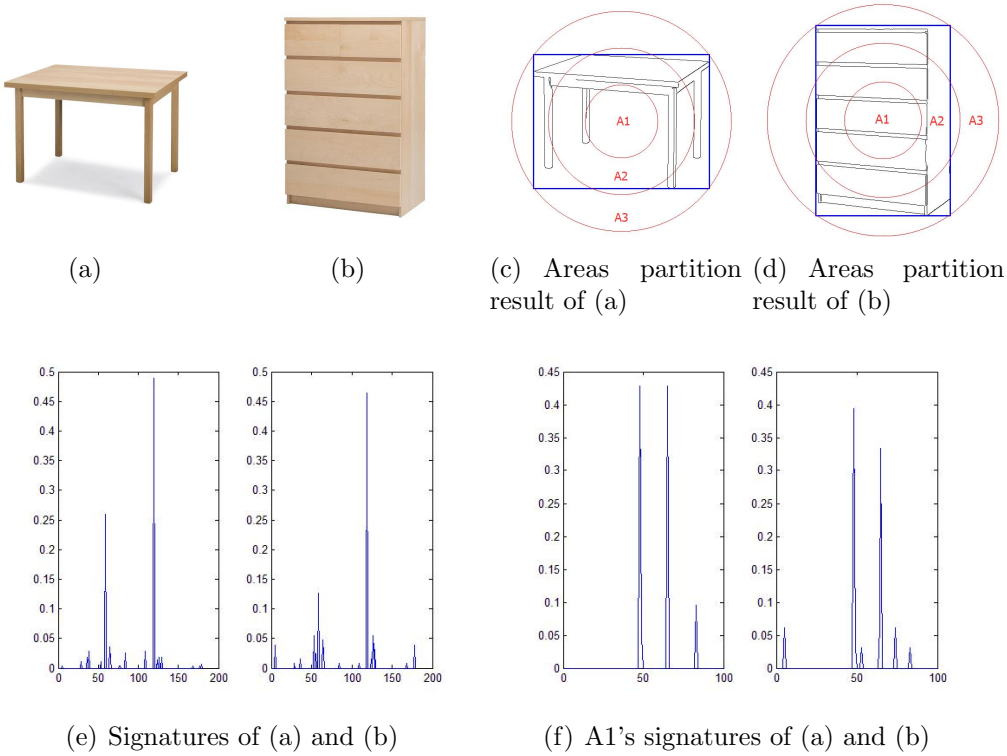
FIGURE 9. Signature generation procedure of BOW

The location information of the image is not factored in the traditional bag-of-words method which leads to the mismatching of the two images. To solve this problem, we divide an image into three circle areas to produce their respective signatures. The specific steps are as follows:

(1) One image is firstly divided into three areas. The edge of the image is surrounded by a bounding box and half of the bounding rectangle diagonal is defined as R. According to the rectangle center O, circles will be drawn by radii of R/3,2R/3,R , respectively. Thus, the three concentric circles divide the image into three areas, A_1,A_2,A_3,as shown in Figures 10(c) and 10(d). The interval distance of concentric circles is related to the value of R, which will be resistant to image scaling attacks. When rotating the image, the three circle areas are relatively invariant, which will thus resist image rotation attacks.

(2) The signatures of the three areas are generated respectively. In the matching process, we firstly use the inner area's signature to obtain a set of candidate images which contain n_1 images. After that, the middle area's signature will be used to select the n_2

images from the n_1 ones. Lastly, a set of n_3 images will be found from the n_2 ones. n_1, n_2, n_3 are the numbers of images and $n_3 < n_2 < n_1$. The values of n_1 ,n_2 and n_3 can be obtained by experience from different databases. In our experiment n_1=50,n_2=30,n_3=8. When a set of images is reduced from n_1 to $n_2$and then to n_3, the most dissimilar images will be removed quickly. Although this operation may slow down the matching speed process, the number of mismatched images can be reduced too. There is a compromise between performance and accuracy. Experimental results show that our algorithm can obtain a set of about 10 images, which contain the original image corresponding to the camera image. Figures 10(a) and 10(b) are two original images and the feature descriptors are very similar as shown in Figure 10(e). The area partition results of them are shown in Figures 10(c) and 10(d). As is shown in Figure 10(f), the signatures of the two images are distinguished just by area A1.



(a) (b) (c) Areas partition result of (a) (d) Areas partition result of (b)

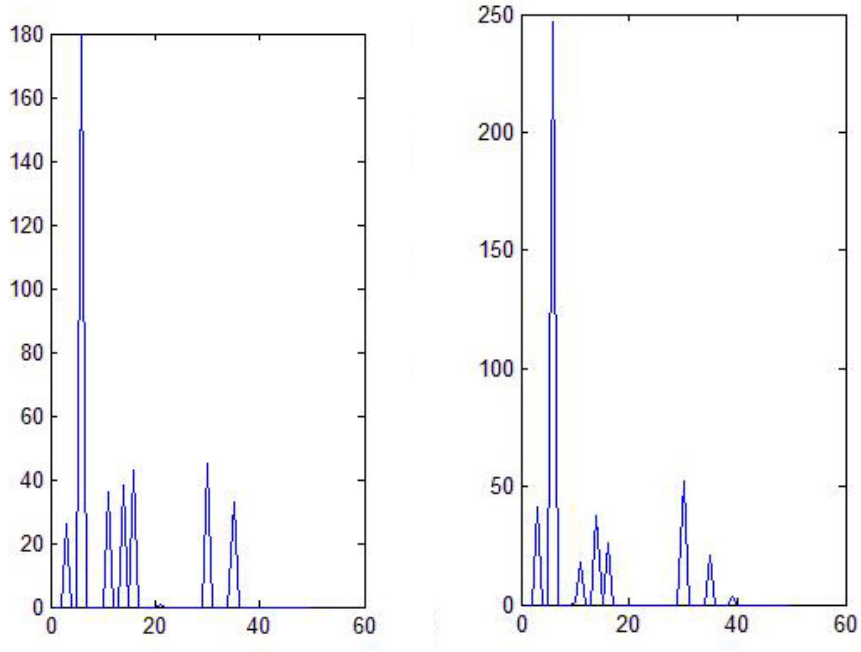
(e) Signatures of (a) and (b) (f) A1's signatures of (a) and (b)

FIGURE 10. Area partition

2.4. **The Proposed Image Signature Matching Algorithm.** The next step is to match the image signature vectors in the database after retrieving the signature of the photo image. The two images are matched when the distance of the two signature vectors is close enough. The Euclidean distance neglects the relationship among dimensions in a signature vector. To solve this problem, we propose a scale transformation invariant to rectify the mismatching caused by Euclidean distance. Experimental results show that this image signature matching method improved matching accuracy.

Assuming that the image signature vector computed based on BOW is x=(x_1,...x_k) , and that there exists scale transformation before and after photographing, so we assume the image signature vector after photographing is x'=(x_1',...x_k'). The relative number of the feature before and after photographing stays roughly the same, that is to say if the number of one feature is more than other features before photographing, the number of the feature will still be more than other feature after photographing. As is shown in

Figure 11, we can see that the signature vector has the linear relationship before and after photographing, and the linear relationship is the scale transformation before and after the photograph is taken.



(a) The signature vector before photographing

(b) The signature vector after photographing

FIGURE 11. Relationship between the signatures

So, the relationship between the image signature vectors before and after photographing approximately equal to be linear which is defined as Formula 1,

$$x' = \alpha x \tag{1}$$

where α is the coefficient which indicates the linear relationship between x' and x . Then an invariant σ is defined as Formula 2,

$$\sigma = \sum_{i=1}^{k} \frac{\sqrt{(x_i - \bar{x})^2}}{\bar{x}} \tag{2}$$

where σ is proved to be invariant before and after photographing by Formula 3. The invariant σ' after photographing is obtained according to the Formula 1.

$$\sigma' = \sum_{i=1}^{k} \frac{\sqrt{(x_i{}' - \bar{x}')^2}}{\bar{x}'} = \sum_{i=1}^{k} \frac{\sqrt{(\alpha x_i - \alpha\bar{x})^2}}{\alpha\bar{x}} = \sum_{i=1}^{k} \frac{\sqrt{(x_i - \bar{x})^2}}{\bar{x}} = \sigma \tag{3}$$

σ' equals to σ prove that there exits a value which is constant before and after photographing. So the invariant can be used to compare signatures to rectify the mismatching results caused by Euclidean distance. After comparing the query signature with the ones in database, a set of similar signatures rank top are selected which contains both the right and wrong results. Then the invariants are computed and compared. The final order of

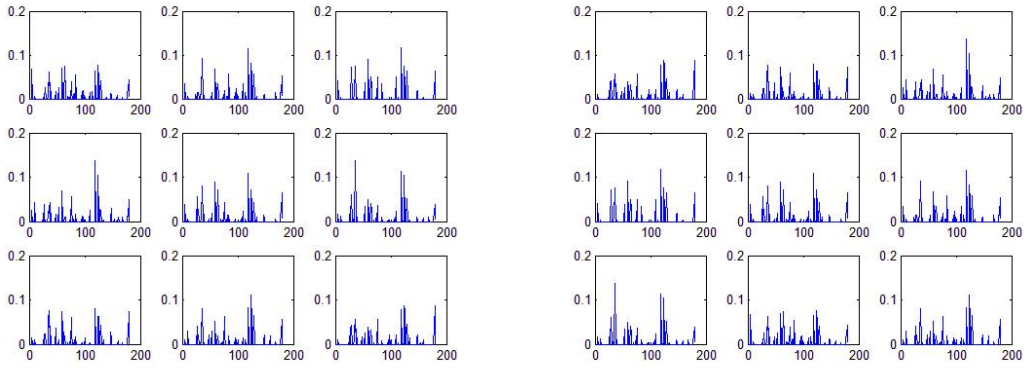
the similar signatures is re-ranked by the invariants comparison results that the order of the right results will be increased and the wrong ones be decreased.

Figure 12(a) shows a set of images retrieved just by using Euclidean distance, whose order starts clockwise from the top left image - from f1 to f8. The middle image is the captured query image. Figure 12(b) is the retrieval results re-ranked by the invariants σ. It shows that the sequence number of f5 ranked from 5 in Figure 12(a) to 1 in Figure 12(b).

Each histogram in Figure 12(c)and 12(d) is a signature of the corresponding image. The horizontal ordinate represents a visual vocabulary which contains 180 words and the vertical coordinates represents the occurrences of visual words.

Table 3 shows the histogram comparison values of the query image with the 8 images in Figure 12(a) by Euclidean distance and σ respectively. The difference among Euclidean distances is very small but the difference of σ is larger.

(a) Euclidean distance comparison result (b) Reordered result by σ



(c) Corresponding histograms of images in (a) (d) Corresponding histograms of images in (b)

FIGURE 12. Comparison of Euclidean distance and reordered result by σ

2.5. **Fine Matching Process.** After rough matching process, a set of similar images as candidate images are obtained. Next, the fine matching method is applied in order to get the final result. At which point, the fine matching algorithm is carried out by ORB.

TABLE 3. Histogram comparison value

	f1	f2	f3	f4	f5	f6	f7	f8
Euclidean distance	0.0 984	0.0 994	0.0 997	0.1 090	0.1 095	0.1 133	0.1 143	0.1 163
comparison of σ	5.17	3.61	19.18	7.65	1.6	4.43	1.99	3.41

Firstly, we extract feature points from both the query and candidate images using oFast, respectively. Secondly, the feature points are described by rBRIEF. Thirdly, the candidate matching point-pairs are obtained by way of cross-match method. The number of the matched point-pairs is called cross-match numbers. Finally, the geometric verification step eliminates matches with wrong feature locations by changing the viewing position. The correct matching points are called inliers. The ratio of inliers number and cross-match number is a key measurement of match result - the larger the better. The detailed flowchart is described in Figure 13.

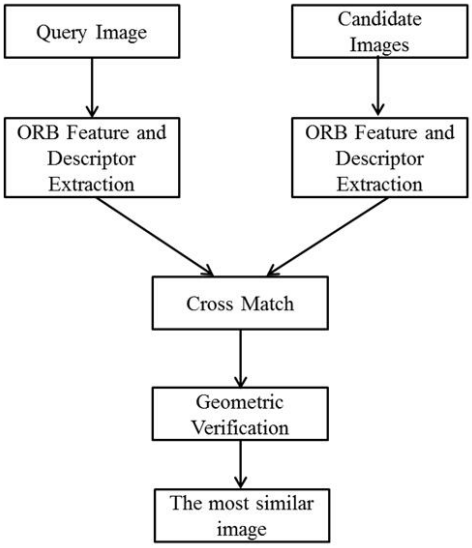


FIGURE 13. Flowchart of fine match

FIGURE 14. A set of candidate images

Through the operation of the previous sections, one result obtained by rough matching is shown in Figure 14. Among them, the middle image is the query image, which is surrounded by eight candidate images ranked from f1 to f8. The query image is matched with eight candidate images using fine matching. Table 4 shows the matching results. By setting the threshold of inliers to 100, and the ratio of inliers and cross-match numbers to 70%, candidate image f2 is selected as the final search result. Experimental results show that fine matching can obtain the most similar image from the rough matching result precisely.

3. **Experimental Results.** Experiments are conducted with the database of thousands of furniture images collected from the IKEA. The visual vocabulary and signatures of the images in database is computed and preserved in the database establishment stage. The visual vocabulary size is set to 180, so every signature of images in database is a histogram of 180 dimensions.

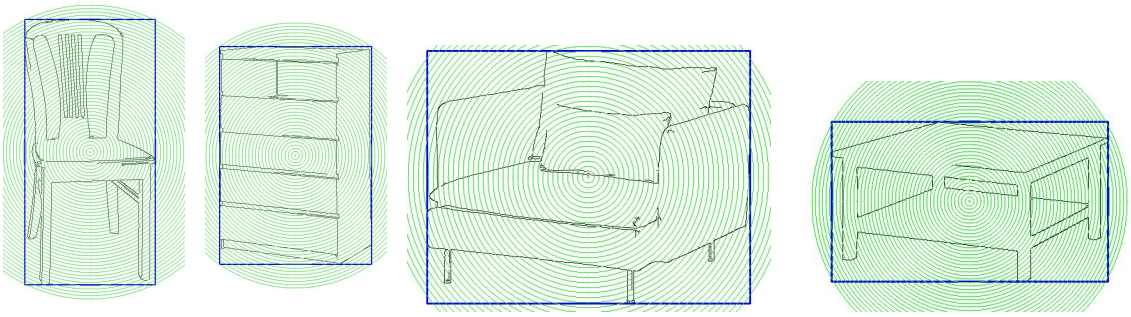
TABLE 4. Data of fine matching

Candidate Image	f1	f2	f3	f4	f5	f6	f7	f8
cross-match numbers	106	218	187	91	182	161	97	116
Inliers numbers	1	171	92	8	87	53	7	13
ratio	0.001%	78.4%	49.2%	0.09%	47.8%	32.9%	7.22%	11.2%

Four groups of experiments are chosen to test the rough matching algorithm in the following. Figure 15 shows four query images captured by mobile phone: Chair, Drawer, Sofa and Desk. For each camera image, we firstly extracted its feature points using proposed concentric circles-based feature points extraction algorithm, as shown in Figure 16. Then, the histogram of signatures will be generated using the preserved visual vocabulary. After that, the signature of each camera image is compared with the preserved signatures in database using the proposed image signature matching algorithm. Finally the first eight retrieval results of each camera image are shown in Figure 17 which proves that the rough matching can ensure that the most similar image to be found in the group of similar images.

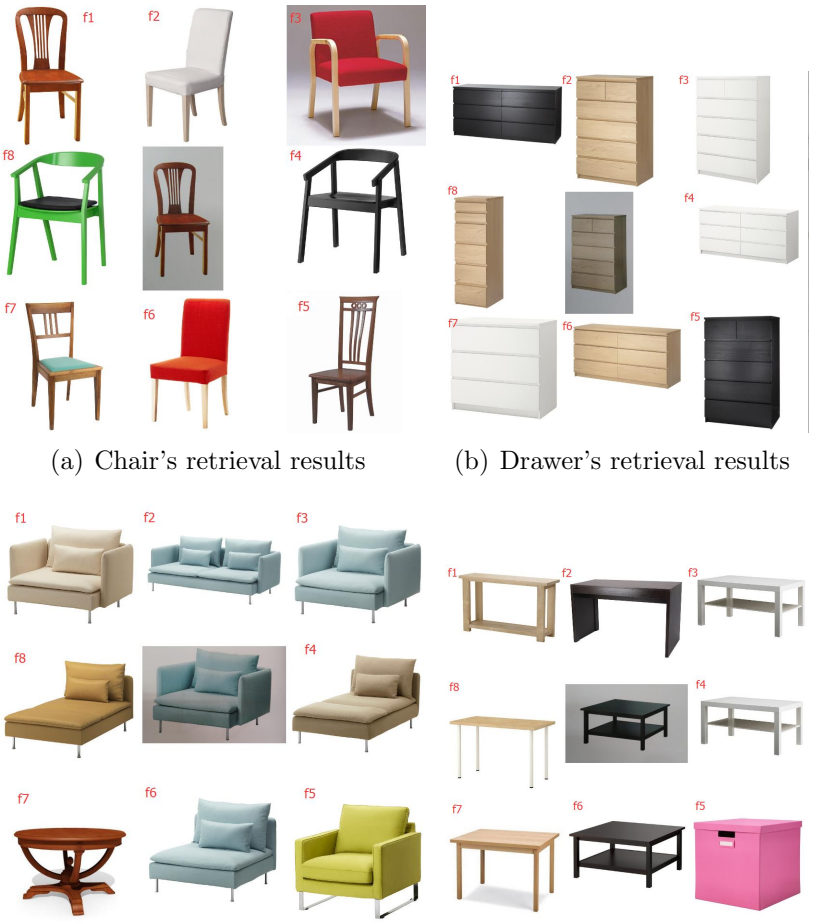
(a) Chair (b) Drawer (c) Sofa (d) Desk

FIGURE 15. Camera images



(a) Chair's feature extraction (b) Drawer's feature extraction (c) Sofa's feature extraction (d) Desk's feature extraction

FIGURE 16. The contour feature points extraction



(a) Chair's retrieval results (b) Drawer's retrieval results

(c) Sofa's retrieval results (d) Desk's retrieval results

FIGURE 17. The rough matching results

The fine matching process is conducted to each group of rough matching results in Figure 17. Table 5 shows the ratio of inliers numbers and cross-match numbers of the four rough matching results. The larger ratio is the better and the best ratio is shown in boldface. So the finally retrieval results of the four camera images are Chair: f1 in Figure 17(a); Drawer: f2 in Figure 17(b); Sofa: f3 in Figure 17(c); Desk: f6 in Figure 17(d). The fine matching results prove that the most similar image is selected from the rough matching results precisely.

TABLE 5. The ratio of inliers and cross-match numbers

Candidate Image	f1	f2	f3	f4	f5	f6	f7	f8
Chair	**90%**	17%	11%	13%	13%	0.9%	10%	0.9%
Drawer	23%	**82%**	24%	23%	31%	35%	38%	34%
Sofa	29%	26%	**89%**	21%	0.9%	26%	14%	27%
Sofa	20%	27%	18%	21%	30%	**70%**	22%	35%

To evaluate our retrieval algorithm, Mean reciprocal rank (MRR) [13] is used as our evaluation metric which considers whether the original database image is retrieved and

TABLE 6. MRR example

Query	Results	Correct Response	Rank	Reciprocal Rank
red chair	black chair, blue chair, **red chair**	red chair	3	1/3
black desk	red desk, **black desk**, blue desk	black desk	2	1/2
white sofa	**white sofa**, black sofa, blue sofa	white sofa	1	1

ranks on the top. The MRR is the average of the reciprocal ranks of results for a sample of queries Q:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{4}$$

For example, suppose we have the following three sample queries for a retrieval algorithm to retrieve the true color furniture of the query sample. In each query, the algorithm returns three results, with the first one being the one it thinks is most likely correct, as is shown in Table 6. Given those three samples, we could calculate the mean reciprocal rank as $(1/3 + 1/2 + 1)/3 = 11/18$ or about 0.61.

Fifteen groups of experiments are conducted to test and verify the validity of our algorithm and each group contains 100 query images. For each query image in a group, the MRR is calculated and then averaged for all queries in this group. As shown in figure 18, we compare the MRR of ours with Eitz's [8], Horace's[14] and Li Fei-Fei's [15]. It shows that our method of combing the rough matching and fine matching gets the best precision.

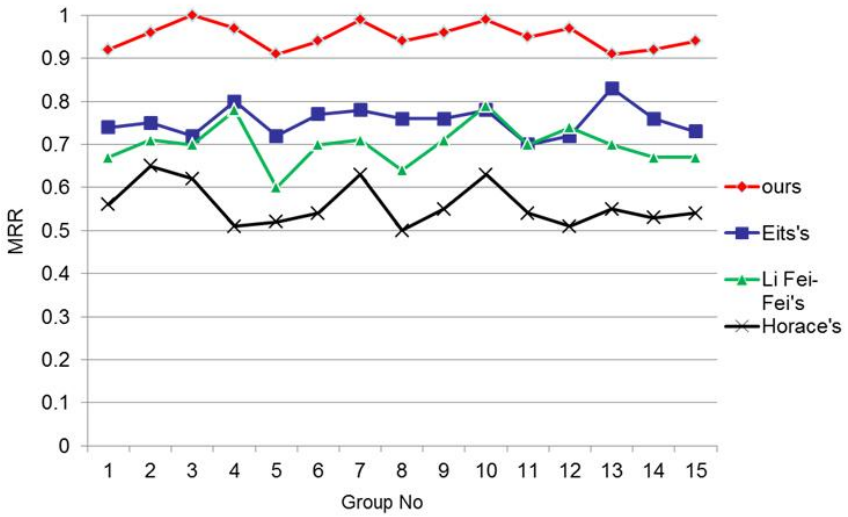


FIGURE 18. Performance comparison of traditional BOW and ours

The time consumption of our retrieval algorithm includes two parts - the rough matching time (RMT) and the fine matching time (FMT). As is shown in figure 19, the rough matching time is that one query image compared with more than 1000 images in database. The fine matching time is one query image compared with 8 candidates obtained from rough match using ORB. The overall time consumption of our algorithm is about 0.7 seconds, but to match 1000 images just using ORB will consuming about 500 seconds. In conclusion, our algorithm can get better performance with less time consumption.

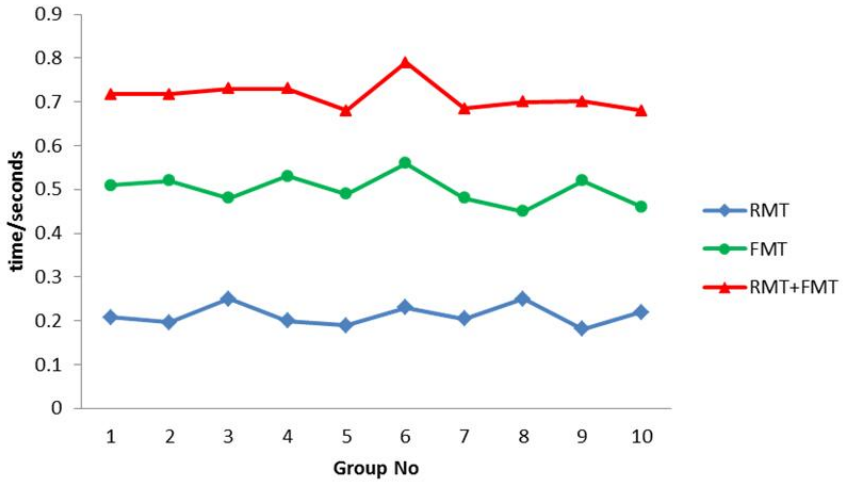


FIGURE 19. Time consumption of our algorithm

We have test our algorithm on a Intel(R) core(TM) i5 CPU 2.80GHz machine with 2G RAM and compare the time consumption of our retrieval system with Eitz's[8], Horace's[14] and Li Fei-Fei's [15]. As is shown in figure 20, our retrieval system performs better.

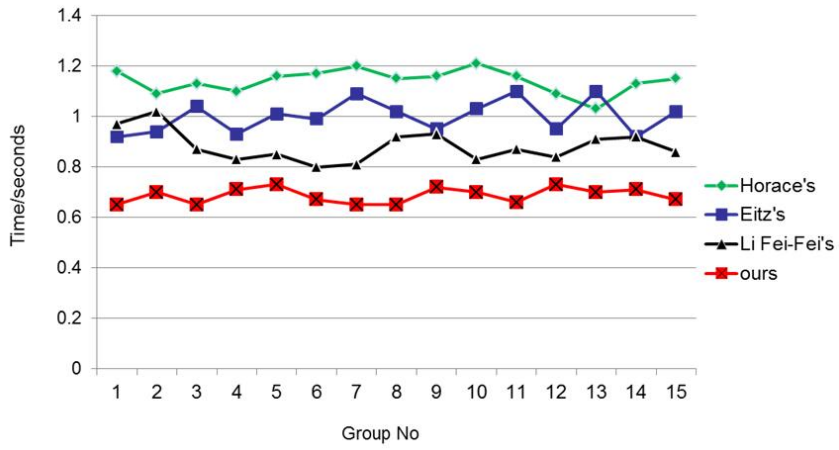


FIGURE 20. Comparison of time consumption

Conclusion. In this paper, an image retrieval algorithm using rough matching and fine matching is proposed for furniture images captured by mobile devices. Firstly, the traditional BOW method is improved by adding a spatial relationship. Secondly, a feature point extraction method on concentric circles is proposed. Thirdly, the GALIF descriptor is improved to have better performance. Finally, a new image signature matching strategy is proposed to improve retrieval accuracy. The experiment result shows our algorithm can retrieve furniture images captured by mobile phone with better efficiency and performance.

Acknowledgements. This work was partially supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (No. 2012BAH91F03) and National Natural Science Foundation of China (No.61370218).

REFERENCES

[1] D. G. Lowe, Distinctive image features from scale-invariant keypoints, in *International Journal of Computer Vision*, pp. 91–110, 2004.

[2] C. Harris and M. Stephens, A combined corner and edge detector., in *Alvey vision conference*, vol. 15, p. 50, Citeseer, 1988.

[3] E. Rosten and T. Drummond, Machine learning for high-speed corner detection, in *Computer Vision–ECCV 2006*, pp. 430–443, Springer, 2006.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, Surf: Speeded up robust features, in *Computer vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[5] D. Gabor, Theory of communication. part 1: The analysis of information, *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, vol. 93, no. 26, pp. 429–441, 1946.

[6] D. Dunn and W. E. Higgins, Optimal gabor filters for texture segmentation, *Image Processing, IEEE Transactions on*, vol. 4, no. 7, pp. 947–964, 1995.

[7] L.-L. Huang, A. Shimizu, and H. Kobatake, Robust face detection using gabor filter features, *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1641–1649, 2005.

[8] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, Sketch-based shape retrieval., *ACM Trans. Graph.*, vol. 31, no. 4, pp. 31–1, 2012.

[9] J. Sivic and A. Zisserman, Video google: A text retrieval approach to object matching in videos, in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1470–1477, IEEE, 2003.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, Orb: an efficient alternative to sift or surf, in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[11] P. L. Rosin, Measuring corner properties, *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, Brief: Binary robust independent elementary features, *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[13] E. M. Voorhees *et al.*, The trec-8 question answering track report., in *Trec*, vol. 99, pp. 77–82, 1999.

[14] H. H. Ip, A. K. Cheng, W. Y. Wong, and J. Feng, Affine-invariant sketch-based retrieval of images, in *Computer Graphics International 2001. Proceedings*, pp. 55–61, IEEE, 2001.

[15] L. Fei-Fei and P. Perona, A bayesian hierarchical model for learning natural scene categories, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 524–531, IEEE, 2005.